



## THE EDITOR'S GODZILLA

-by Lenard R. Roach

### COMMODORE EXORCISM

A simple religious ceremony ended up as a big party featuring the couple of Commodores located in my home (nicknamed Limehouse due to the puke green paint on the living room walls.)

For years, my old home had suffered problems with plumbing, air conditioning, heating, and so forth.

As the calendar flipped from 2022 to 2023, I took a week off from my day job and started to work on making a particular room fit for occupancy, in case I had friends over who needed lodging for a night. Yet, the particular room had an eerie feel to it, like something was left behind that needed to be thrown out and burned with torches. I went through the closet and found nothing but empty hangers on a wooden rod. The room was completely empty except for a Kansas City Chiefs trash can and a puzzle picture of Jesus hanging on the northwest corner wall. Even my cats, Lee and Tot, could feel there was something amiss and only entered the room when I went in; otherwise, they stayed out even though the door was open in case they wanted to explore while I was away at work.

Despite the eerie feeling, I began to redecorate the room. From a second-hand store, I bought a new/used twin bed, laid down some pillows, sheets, and covers on it, and moved the bed to the northeast wall. I put in a computer desk and chair—hand-me-downs from a friend who worked at a hotel where rooms were being remodeled—and placed them in the southeast corner. My sister-in-law, Wynn, made some decorative glass angels, and I placed them on the shelving above the bed. Even with all this, the room still “felt” wrong. I decided it was time to call in some otherworldly help, but not of the evil kind... quite the opposite in fact. I called my best friend, Jon, and asked that his pastor wife, Barbara, come and perform a ritual of “anointing” the house so it could be dedicated for the service for the forces of truth, justice, and good. We scheduled May 30, 2023, at 1:00 pm, as the time for the ceremony.

When May 30 came, Pastor Barb and her whole family came by to perform the ceremony. With a seven dollar an ounce bottle of anointing oil, Barb first anointed me personally as the owner of the house and prayed that I would have peace in me while going through the redecoration. Next, she anointed the entrance to Limehouse, putting the oil on both door posts and the lintel and prayed again. Pastor Barb did this ritual to all the doors that were in Limehouse, including the bathroom door and the back door leading to the yard. Lee and Tot followed us as this ceremony was going on. Once the bedroom door posts of the particular room were anointed, the cats walked in and immediately started sniffing around the room and jumping on the bed as well as on the few sticks of furniture that were in the room.

Pastors Jon and Barb were about to leave when their grandsons, Jade and Hunter, stepped up to me and asked if they could play on the computers, referring to the Commodore 64 that was hooked up to the big-screen television in the living room and the Commodore 128 in the computer room.

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

“Sure,” I said. “What game would you like to play?”

Jade spoke up first, “Can I play 'Rampage?’”



By a stroke of providence, I had been looking online for a working copy of Rampage a few months ago so that I could play it during the little free time I had. I was blessed to find one and had downloaded it from the web to my C128 and the SD2IEC. Then I had gone through the process of transposing the game out of its .D64 archive and onto a 5.25” floppy disk.

I booted the C64 in the living room and loaded the game. While the Commodore 1541 disk drive was going through the process of loading the game into the computer, I went to the computer closet to dig up a couple of joysticks. This is when Hunter, the youngest brother, asked me if he could play, too.

“Did you want to play Rampage against your brother on the big screen in the living room?” I asked Hunter.

“No, I want to play on that.” Hunter was pointing to the C128 on my computer desk in the computer room. Well, now! I would have both computers running and would have to play catch up between each room to make sure the machines were running well during each game's progress.

From the living room, I got the joysticks for Jade, and he and Grandpa Jon went nuts playing

Rampage on the big screen. Then I quickly turned my attentions to Hunter and the C128. Pastor Barb was with me in the computer room as Hunter was trying to decide the game he wanted to play. Of course, Hunter did not know what I had in the way of games. Back in the computer closet where all my boxed games were stored, I found a copy of each: “Batman: The Caped Crusader” and “Batman: The Movie.” Since I knew that both boys were big fans of Batman, I thought that they would both like to play the games. Hunter got to select first since Jade and Grandpa Jon were presently playing Rampage. Hunter chose The Caped Crusader. I booted the game in 64 mode of the C128, and soon Hunter was off trying to fight crime in the digital world of Batman.

I went back into the living room and interrupted the game play of Rampage. With Batman: The Movie in hand, I showed Jade the game. Jade lit up. “Yeah, I want to play that!” he exclaimed. Rebooting the C64 in the living room, I loaded the game, and soon Jade was fighting his way around Gotham City, beating up criminals while chasing down The Joker. Both boys were in computer heaven. However, problems soon erupted.

**Scientist added the letters AH to the periodic table. It is the element of surprise.**



With no game instruction manuals to tell the boys how to use such things as grappling hooks, the Batarang, and the Bat Computer, they were soon stuck on level one and stayed there. With Hunter being so young (about six years old), he was quickly losing patience with Batman: The Caped Crusader and wanted to shut down the computer and go home so he could play “Minecraft” on his handheld.

“The games were too hard,” Jade said. “I couldn't go anywhere.”

“Same here,” Hunter chimed in.

“Maybe you can come back sometime and try to beat the games,” I said with a smile.

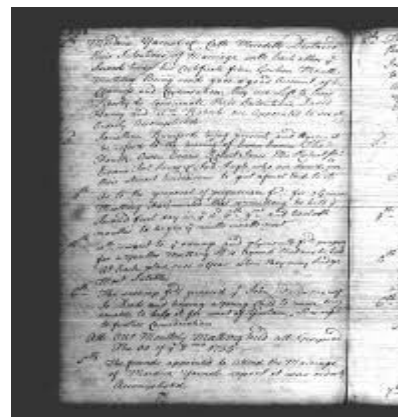
The family filed out of the house and towards their car. Thinking how fun it was for me to share my Commodore computers with the younger generation, I watched them as the parents loaded up their gear, buckled up the kids, climbed in, and drove off. Despite the simplicity of the Commodore, the upcoming generation had difficulty understanding such a system, even when using easy games such as the Batman series. Or was Batman THAT easy?



Batman: The Caped Crusader was still loaded and running on the C128 in the computer room. I sat down at the desk and started maneuvering the character around trying to get out of the Bat Cave. Before long, even I got frustrated and quickly shut off the machine.

Hmm. Maybe the Commodore wasn't quite as simple after all ....

Jade was also facing his own set of frustrations with Batman: The Movie. Even though he could maneuver his Batman character easily around the sewer, he was having troubles finding his way out of the sewer and onto the next level. Before long, Jade too was frustrated and wanted to go home to play on his handheld, something he could understand better than the complexities of a Commodore 64 game. Jon and Barb decided to call it quits, pack up, and head home. I thanked them for coming by and blessing my home. I turned to Jade and Hunter and said, “Thanks for coming by. I hope you had fun on the computers.”



## MONTHLY MEETING REPORTS

Scientist added the letters AH to the periodic table. It is the element of surprise.

-by Robert Bernardo & Dick Estel

## September 2023

Turnout was unusually low for this final summer meeting. When Dick arrived, Robert was nearly done setting up the equipment, and no one else was around. Roger and Dave had both let Robert know they would not be present, and we recalled a period when membership was at an all time low and meetings were sometimes just Robert and Dick.

An hour later Bruce arrived, increasing the crowd by 50%. The pre-meeting conversation ranged far and wide, and included our condolences to Robert on the loss of his mother, who passed away recently at the age of 93.

Robert also noted that he will be attending his 50-year high school reunion soon. This prompted brief discussion of Dick's 50th, 16 years ago.

In a blast from the past, Robert gave everyone a punch card, one of those 3 by 7.5 inch pieces of lightweight cardboard with holes that constituted a code that can be read by a computer. The basic technology dates back to the late 19th century. Dick recalled that a TV station where he worked used these cards to create the daily log of programs, commercials, and announcements. Driven by idle curiosity, Dick determined that there are approximately 800 spots where a hole could be punched on each card.

The first matter of official business was the club "picnic," which is now actually a lunch at the usual time but at a different restaurant. At a club picnic, there is no official business and no demos, and family members are invited. The location chosen for this year is the Andiamo Italian restaurant in Clovis. Also because of conflicts, the

lunch will be on October 8, the second Sunday instead of our usual third Sunday meeting time.

Robert will be the official videographer at AmiWest Show, October 12 through 15 in Sacramento.

Meanwhile, the Interim Computer Festival will take place in Seattle September 30 and October 1. The show is open to all classic computers.

Robert is planning to attend the Bay Area Maker Faire October 20 – 22. Plans are for Duncan MacDougall to join him, although Robert has not yet received an acknowledgment of his request to attend. In any case, Robert will be going to Seattle in late October, and he will visit Ray Carlsen in order to deliver items for repair and pick up items that have been repaired.

Robert had brought the books, "From Vultures to Vampires," volumes 1 and 2. The books told the story of the Amiga computer starting in 1984 and going to present day. Volume 2 even told the story of how the Commodore name eventually landed with a Dutch company. Volume 3 will come out later this year. Bruce expressed his interest in buying his own copies of the books, but when Robert told him the price of each volume, he quickly lost interest.

Moving on to software demos, Robert let us look at rarely seen classics from long ago, the Hayden Software Temperature Lab and the accompanying Light Lab. Besides software, the boxes included sensors, an interface for the sensors, a thermometer, extensive manuals, and a lot of other items. An example of experiments that could be performed with Temperature Lab was to determine how long it takes ice to cool a soda (a rather inexact phrase). The burning question of "How do fireflies give off light?" was also answered.

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

Robert had some disappointing news about his other planned demonstrations. He had brought his Commodore PC20-III DOS machine and hoped to run a PC version of GEOS. The program had installed correctly, and the machine pretended to be booting up this graphical operating system but just displayed the word “Loading” until the user gave up and moved on to something else. Robert had also brought the Mega65 (C65 clone) and was going to give a presentation on GEOS 65, the GEOS version which was originally developed from GEOS 128 and was now converted to run on the Mega65. Unfortunately, though Robert had the microSD card with the GEOS 65, the Mega65 did not recognize the card as being compatible with its system. (Robert discovered later that he had to prep the microSD card with the Mega65, before using it with the system. Live and learn!)

## October 2023



### C= COMPUTERS REPAIRED

-by Robert Bernardo

Hurrah! Duncan M. of The Other Group of Amigoids (San Jose, CA) repaired two computers for me. First was the Commodore PC20-III, a computer which lay unused and unrepaired for years. He diagnosed that the mechanical hard

drive had frozen, but it did boot up from DOS floppy disk. He replaced the old HD with a Compactflash card, giving it DOS 6.0. Then he upgraded other components of the computer—maxing out the memory, plugging in a more efficient NEC V20 CPU, adding a 3 1/2” floppy disk drive alongside the standard 5 1/2” floppy drive, building and installing a Snarker Barker sound card (Sound Blaster clone) with 3D-printed volume control wheel, and putting in a RGBI-to-VGA video card. On the CF card, he installed a slew of PC games, especially games that used sound through the sound card. He admitted that he hadn’t finished rebuilding a CH (Kraft) joystick for me to use on the PC (joystick connecting to the port on the sound card), but that if I wanted to use a mouse on the computer’s mouse port, I could use an Amiga mouse.

As an old PC, it is slow to boot up, and the CGA games are not really that colorful. I’m thinking of installing GeoWorks onto it, so I can get that GEOS desktop feeling.

The second, repaired computer was one of my Amiga 3000’s, the one that I’ve had since the early 2000’s. I had bought it for \$25 from a former TOGA member. It didn’t boot up, and Duncan had to repair it back then. It had NiCad battery damage, causing a small crater in the motherboard. Duncan snipped out the old battery, cleaned off the board, and put in a 2 gig mechanical hard drive with Amiga OS 3.1. I had used that computer for years, and now it was acting up, giving me a black screen and not even showing an early start-up menu.

This time he disassembled the computer and carefully looked at the board. He noticed that the Agnus socket was not in good condition, something he missed years ago. With some effort, he desoldered the old socket and installed a new socket. Also when he tried to dump the old hard drive via a Zulu SCSI onto a CF card, he

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

discovered the HD had a lot of read errors. So, no use in trying to archive that drive, and he did a new install of OS 3.1 onto a CF card, i.e., no more mechanical hard drive but a CF card on the Zulu SCSI board. With the new install, he put in a complete set of WHDLoad games. With those repairs done, he tried to boot the computer, and it worked!

Duncan noticed I had brought a boxed Spectrum 24-bit video card and a boxed Ariadne-II network card. After we discussed whether the cards should go in the A3000 or my currently non-working A4000, I decided they should go into something working, the A3000. The Spectrum card went in relatively easily, and instead of using the CyberGraphX v2.16 disks that came with it, he used a newer (latest) version of the CybergraphX software. Software installation was straight forward... no problems. He attached the pass-through cable from the A3000 VGA port to the Spectrum's pass-through port; now applications which could not use the 24-bit desktop would be passed through to a compliant multi-scan monitor.

He powered up the computer, the CyberGraphX logo was displayed, and his monitor came up with a 24-bit screen! He adjusted the settings for 800 x 600 and then tried to find a good-looking JPEG image to use on the desktop. Unfortunately, the JPEG's that he tried took a long time to load (slowness I attribute to the A3000 only having the standard 68030 at 25 MHz., though it had the new CF card drive and a 256 meg ZorRAM). Instead of JPEG's, he advised that I look for nice IFF images to put on the desktop. Speaking of loading, the A3000 took longer to boot up, because it had to recognize the video card and the ZorRAM in the system. The benefits were the 16-million color desktop, the freeing-up of valuable Chip RAM, and the scads of Fast RAM.

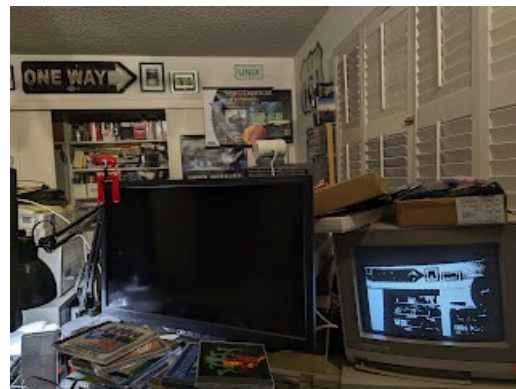
He installed the Ariadne-II network card, having to use a clamp to adjust its mounting bracket to align more squarely with the slot in the A3000. He left it up to me to install the software on disk.

We discussed OS 3.9 (which used more colorful icons and which would be good for the look of the 24-bit desktop), but I was leery of using it on a '030/25 Mhz. Amiga and not on a faster machine. He put forward the ideas of using OS 3.2.2 (which didn't necessarily require a faster computer) or a BF9060 accelerator board (a thousand dollars with 68060 CPU!). I didn't want to go the OS 3.2.2 route, and I didn't want to go the '060 accelerator route (but when is the Apollo Kraken accelerator going to come out?!). Instead of further modding the A3000, we agreed that the Amiga 4000 should be next to be repaired.

Look for the Commodore PC20-III and the Amiga 3000 to appear at future meetings of the Fresno Commodore User Group and the Southern California Commodore & Amiga Network!

## LIVECAM THE C128 VDC + COMPUTEREYES

-by Guest Contributor Cameron Kaiser



If we're going to make the little old 8-bit MOS 6502 into Skynet — because we already know

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

what the Terminator T-800 CPU is — then it's gonna need to see. How can it exterminate the last remnants of humanity without vision?

And we'll use something period-correct, too. While our favourite Cyberdyne Systems Model 101 was busy stalking Sarah Connor in 1984, the product it might have (slowly) viewed the world with was already on the market -- the Digital Vision ComputerEyes. Check out the little beige camera perched on a stack of disk boxes, attentively surveilling the room at just a few, uh, seconds per frame as displayed on the monitor. Plug in a composite video source, connect it up to your Apple II, Commodore 64 or (in 1985) Atari 8-bit, and wait about six or seven seconds to identify targets — or almost fifty [seconds] for the highest quality. If Skynet had chosen this option we might never have had Judgment Day.

The slow capture speed meant it was never intended as something to view live, and on the Commodore and Atari versions, DMA interference meant you could *only* capture with the screen off which would seem to make any live-ish feed impossible. But the Commodore 128 has a *second* video chip that *doesn't* interfere. Let's turn the Commodore 128 into a really slow potato-quality live camera you can interactively watch and freeze-frame — and then, in exchange for 11% of the screen, make it capture almost 25% faster! Time-lapse video proof at the end!



Digital Vision was founded by David Pratt in March 1984 in Needham, Massachusetts to commercialize his Apple II slow-scan video

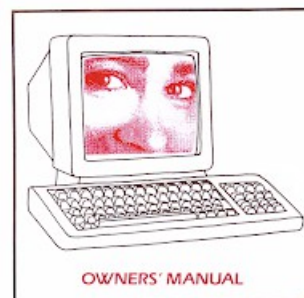
capture board. Pratt's design used minimal hardware, shunted most of the work to the 6502 CPU, and existed as an external box passing image data over the game port rather than a card for the internal slots. It captured at the standard Apple II HGR resolution of 280x192. The package included several demonstration images of the long multi-level capture mode, such as "Nan," a picture of their office manager's cousin (above, here from the higher resolution Commodore version). Named ComputerEyes as a desperate pun, it hit the market in 1984 at \$130 for the box alone [in 2023 dollars about \$380], or with a high-quality NTSC bullet camera for \$350 [\$1020], easily the cheapest such solution available. Source code was helpfully provided, something we'll take advantage of presently, as well as a later image conversion and enhancement package for \$25 [\$70].

## COMPUTEREYES™

By

■■■■■■■■■■ DIGITAL VISION ■■■■■■■■■■

VIDEO ACQUISITION SYSTEM  
FOR  
COMMODORE® COMPUTERS



The product was well-reviewed despite the slow performance, but the best-selling computer of 1984 was the Commodore 64. One of Digital Vision's distributors prevailed upon them to develop a C64 version, which could generate a dithered VIC-II high-res image of 320x200. Hitting the market in the fall, it sold for the same price as the Apple version and also included the assembly source for the driver, alongside additional utility disks to use it with KoalaPainter, Doodle, Print Shop and Flexidraw at \$15 [\$40] a pop.

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

COMPUTE!'s Gazette in June 1985 complained that tweaking it for the video source took a lot of trial and error and echoed that the time required to take a picture was "really frustrating," but editor Charles Brannon was ultimately impressed with the results, saying, "Pictures justify themselves. It's just plain fun." The picture on the manual cover reportedly came from a Pepsodent advertisement.

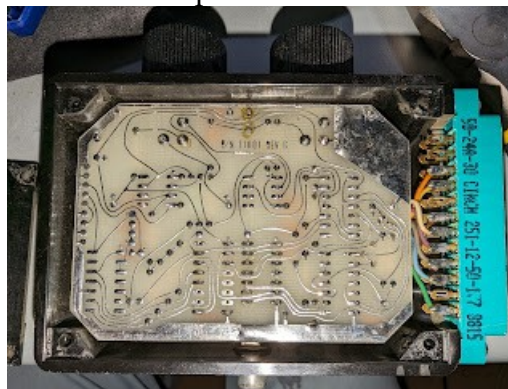


Here are my own two ComputerEyes. The one on the left is older but they appear to have the same basic hardware, indeed the same basic hardware as the Apple II version they descend from. The unit requires you to manually adjust synchronization with the connected video source as well as the brightness of the signal using the potentiometers on the side; the single composite RCA jack is on the other side. A separate PAL version is mentioned in the manual but I've never seen one, and this article concentrates exclusively on the NTSC version.



The device connects to the user port where it is controlled by reading and writing the user port pins directly, again in nearly the same fashion as it would have worked with the game port of the Apple II. Here it is plugged into my recently repaired "daily vintage driver" Commodore 128DCR, where it's somewhat of a tight fit against the power and VDC RGB cables.

Sales were strong enough to justify a port to the Atari 8-bit family in 1985, which was probably the best of the three due to the Atari being able to generate a very credible multi-level greyscale image. By contrast the original Apple II version was strictly monochrome, and the Commodore port could only generate Koala-style multicolour greyscale through post-processing (and an extra \$15). Later the Apple hardware was developed into a faster internal card to support 560x192 super high res and multiple greys, as well as an Apple IIgs version at 320x200 with up to 16 colours or levels of grey, but the Commodore version was never updated.



The strict requirements on timing were a consequence of how little hardware was actually inside the box. The bottom of the Commodore 64 unit has no screws and is held on by contact cement, so you just pry it up (though I stopped short of yanking the knobs off and loosening the collar on the RCA jack to turn the board over, since I don't want to run the risk of breaking anything). There are a handful of what looks like small logic chips (likely 74LS series, based on what's on the Apple II version) and a few discrete

**Scientist added the letters AH to the periodic table. It is the element of surprise.**



components, and most of the user port lines aren't even connected. In this view user port pin A is at the bottom right of the image, so we only have connections to A (ground), CDEFH (data), KL (data) and 2 (+5V, which powers the unit). None of this would be able to support a high-bandwidth link.

However, diving through the source code demonstrates the hardware is even more constrained than you think. A 59.94fps (interlaced to 29.97fps) NTSC video frame lasts 16.68 milliseconds, heralded by a vertical sync pulse, after which 262.5 lines of 63.5 microseconds apiece follow, each of which starting with a shorter horizontal sync. No one would expect a ~1MHz 6502 to be able to keep up with that beam; an NTSC C64's system clock runs at exactly 315/308MHz, or 1022727.2 cycles/second, or 0.977778 microseconds per cycle. The fastest an NTSC 64 could acquire data would be seven cycles per sample (an absolute load from the CIA chip and a zero page store), at this speed a little under seven microseconds and barely time for just nine captures of an entire scanline. Even if the Atari version got all 1.79MHz of its own clock speed, which it didn't, it could manage maybe fifteen. (The Atari 2600's 6507 gets away with making its programmers "race the beam" due to lower resolution graphics and the requirement that you load some of the TIA's data in advance, such as during blanking periods.)

But the ComputerEyes can't race the beam either. For that matter, it can't even distinguish between the two types of sync; you have to check how long the sync pulse is present and decide yourself if it's vsync or hsync. And then there's the acquisition process:

```
*
* ACQUIRE ONE COLUMN - WAIT FOR VSYNC
PLUS SOME HSYNCS
*
```

```
CLP JSR KBTEST ;TEST FOR KEYPRESS
BNE ADONE ;ABORT IF SO
JSR WVERT
CLC
PAGE
*
* THE INNER LOOP - GET DATA AND SAVE
*
ILP LDA TABLO,X ;GET HI-RES PTR FROM
TABLE
ADC XOFF ;ADD X OFFSET
STA PTR
LDA TABHI,X
BEQ WNXT ;IF 0, DONE WITH COLUMN
ADC XOFF+1
STA PTR+1
*
WHLPL BIT CEREG ;WAIT FOR HOR SYNC
BMI WHLP1
*
BVC AOV0 ;TEST VIDEO DATA
*
LDA BITPOS ;DATA = 1: SET BIT
ORA (PTR),Y ;
BCC OV0
*
AOV0 LDA BITPOS ;DATA = 0: CLEAR BIT
EOR #$FF ;COMPLEMENT
AND (PTR),Y ;MASK OFF BIT TO CLEAR
*
OV0 STA (PTR),Y ;RESTORE
INX ;NEXT ROW
BNE ILP ;GOTO TOP OF INNER LOOP
*
* COLUMN DONE - UPDATE BIT POSITION,
CHECK FOR LAST COLUMN
*
WNXT LSR BITPOS ;SHIFT BIT RIGHT
BNE CLP ;NEXT COLUMN IF NOT SHIFTED OUT
```

This block of code from the Commodore ComputerEyes driver captures an 8-pixel-wide vertical strip (due to the odd way Commodore high-res graphics are organized) and is called in a loop to draw the entire screen from left to right. The ComputerEyes' interface is accessible through CIA #2's data port register B at \$dd01 (here called CEREG), which is wired to pins CDEFHJKL on the user port. Pin L is where sync is detected (bit 7 in the register) and pin K is where the beam is sensed (bit 6). The routine

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

WVERT waits for vertical sync, and the loop at WHLP1 waits for the horizontal sync, using the very useful BIT instruction which ANDs the accumulator to set the Z flag plus loads bit 7 (sync) into the N flag and bit 6 (video) into the V flag, all in one go. But wait a minute: it checks the video data (BVC AOV0) immediately after the horizontal sync signal stops. There is no pause or wait for the electron beam to get to a certain point across the screen. Wouldn't that mean you'd get the same vertical line of samples at the same place each time?

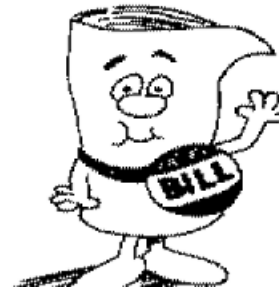
The answer, surprisingly, is no. On each succeeding frame an internal counter stalls latching the sync and sample bits one dot width further along the scanline each time. This is why scanning takes so long: sampling each vertical stripe of pixels requires at least one NTSC frame in duration and every turn of the routine waits for vertical sync before acquiring the next one. Do the math and you get a minimum interval of 5,334 milliseconds to read the entire screen — that's five seconds and change; remember that number — plus any additional setup overhead to equal the measured wall-clock time. Since the 6502 isn't fast enough to sample the scanline to begin with, the hardware takes only a single sample per scanline and moves the sample point along the scanline internally. Even if you try to chase the beam yourself, you'll just get a row of the same data because it never does any other sampling for the rest of a given scanline. (Believe me, I tried.)

The overhead of the driver's initialization code adds up somewhat, too. With a stopwatch my NTSC system took a hair over seven seconds from start to visible image. We'll come back to this section of code later on.

You'll have also noticed that you're only getting one bit of video data, so how does it get the shading information to do dithering or greyscale? Simple: it runs multiple passes with different

sensitivity levels. Earlier in the file the driver sets the Data Direction Register for register B so that only the upper two bits (6 and 7) are inputs from the ComputerEyes, and the lower six are outputs to it. Of the lower six, however, pin J is not connected, so bit 5 is not used, and the software does not appear to use bit 4 either. Instead, bit 3 is used to turn on and off the sampler logic, and bits 2-0 set the sensitivity. The "fast" scan stores a 7 here and relies entirely on analogue wobble to dither, but the 50-second highest quality scan uses all eight possible values and merges the output. This "slow" mode is what took the picture of "Nan."

The capture system also doesn't get the entire frame, only the middle 200 lines of it (192 for the Apple and Atari). This is a "fast" capture I did off my paused DVD of this frame from Schoolhouse Rock's "I'm Just A Bill."



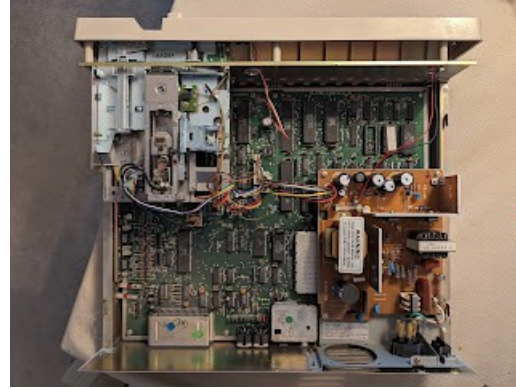
It's a good choice for this, actually, since it's high contrast line art and the box renders it well even at just the default sensitivity, but chops off the top and bottom of the picture.

**Scientist added the letters AH to the periodic table. It is the element of surprise.**



That said, *because* of those technical limitations, the device was cheap enough for retail sale instead of being some obscenely expensive frame buffer only professionals could afford. But the software was doing so much of the work that the timing couldn't tolerate even small interruptions. This wasn't a major problem on the Apple II but the C64's VIC-II video chip periodically stalls out the CPU to grab display data over the bus, referred to as "badlines," which would cause the computer to go out of synchronization with the capture process. This is best demonstrated in the software's own brightness adjustment mode shown here, where the screen must be on to see the pixels so you can tweak the brightness dial, but when the VIC-II asserts itself the resulting scan will end up warped and distorted. For a stable image the screen must be turned off during acquisition, something also required by the Atari version due to its own video DMA.

That brings us to the Commodore 128 and the MOS 8563/8568 Video Display Controller, its second video chip.



The 128's two video chips sit in individual shielded "cages." On this view of the 128DCR, the enhanced VIC-IIe's cage is peeping out from the west edge of the power supply, and the VDC's cage is under the center of the power supply. Originally designed for the scuttled Commodore 900 Coherent "Unix" workstation, the VDC was repurposed to provide the 128 with a true 80-column digital RGBI display for business applications and CP/M, but its poor initial yields and notorious quirks repeatedly bedeviled the development team.

Unlike the VIC-II and VIC-IIe, which as integral parts of the system bus, can directly read shared memory and stall out the CPU when needed for DMA, the VDC is a standalone chip with its own separate video RAM, either a baseline 16K in the flat 128 and plastic 128D, or 64K in the 128DCR, which also had an upgraded 8568 instead of the O.G. 8563. Also unlike the VIC-II(e), the VDC is fully programmable; you can set up many different types of displays with different dimensions, memory requirements and flexible layouts, even down to scanlines and blanking intervals.

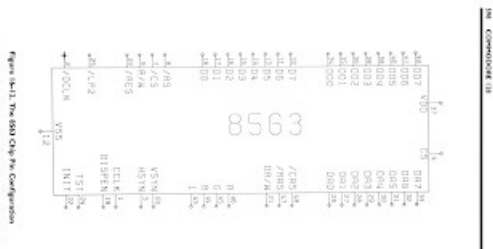
Officially the chip was only used by the Commodore 128 Kernal in text mode, and an initially undocumented 640x200 high-resolution mode existed that was exploited by GEOS 128, BASIC 8, and other tools, but it's possible to create interlaced screens far out of NTSC or PAL

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

spec, such as at least one demo [VDC Mode Mania V1.11 at

<http://csdb.dk/getinternalfile.php/163569/vdcmode/mania-v111.zip>

that can generate an 800x600 monochrome image some monitors will unwillingly display, or a more useful multicolour 640x480 mode. The VDC uses a different 16-colour RGBI palette and generates (at least on the 60Hz NTSC 128) a CGA-compatible display, though one of the pins also provides a composite monochrome signal for non-RGB monitors. Unlike the rest of the system which is ultimately derived from a 315/22MHz (i.e., 14.31818MHz) crystal, the VDC originally ran at 16MHz as a holdover from its original role in the CBM 900, requiring the VIC's 8.18MHz dotclock to be doubled and fed to it in lieu of the original signal.



Accessing its registers and memory has similarities to the Texas Instruments TMS9918 video chip family, which also has its own independent RAM — in fact, on systems like the TI 99/4A and Tomy Tutor, that's practically all the RAM it's got modulo the CPU scratchpad — in that only a single pair of memory ports is exposed directly to the processor, through which all chip registers and RAM are accessed. The VDC will even auto-increment its memory pointer on reads and writes, like the 9918, and provides a crude sort of blitter for local copies and memory wipes. But with the VDC, those two ports are the only view into the chip's running state. The 9918 can generate an interrupt when the screen is updated; the 8563 doesn't do that (the 8568 can, but the interrupt line isn't connected). This is particularly important for the VDC, because it can take a

variable amount of time to generate a frame depending on how it's configured, but the CPU has no choice but to poll the VDC's status register to determine if it's ready or not before asking it to do something else. When the VDC is busy, accesses are simply ignored, making rapidfire updates a matter of risky timing. On top of that, the bus logic was incomplete, so things like indirect memory access instructions that may trigger multiple reads cause the VDC to increment its memory pointer multiple times, overshooting the target address.

But for as finicky a chip as it is, the VDC has two distinct advantages for this application. First, it's still visible when the C128 is in 64 mode (as well as the additional VIC-IIe registers such as 2MHz mode and reading the 128's additional keys), so we can use it with the unmodified ComputerEyes driver and not have to muck with the timing. Second, because it doesn't participate on the bus except for its two ports, it also doesn't do DMA — so we can use the VDC to display the image, because we don't have to turn it off.

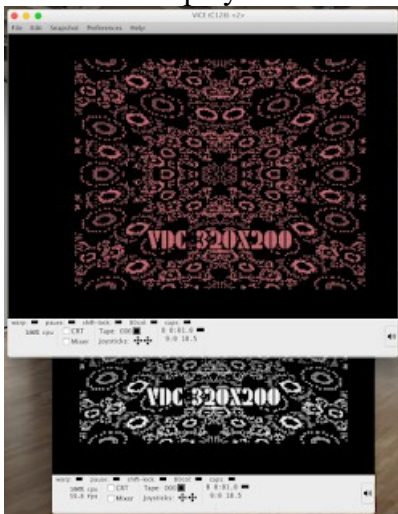
Some of you will be asking why we don't just capture in 2MHz mode, since unlike the VIC-IIe the VDC can still display an image when the CPU is accelerated, and we can double the speed in 64 mode as well. Unfortunately, it's not that simple. "2MHz" mode is implemented by doubling the system clock, so the actual *maximum* speed of an NTSC 128 is precisely double that of an NTSC 64, i.e., 315/154MHz, or 2.0455MHz. I say maximum because not all the parts in the 128, in particular the SID and (relevant to us) the CIAs, are rated at the faster speed. When the CPU accesses these slower chips while running doubled, the VIC-IIe "stretches" the clock, potentially inserting an extra cycle if the access doesn't match the slower clock rate. This means an unpredictable portion of these I/O accesses, and the instructions that make them, will run at the slower speed with an additional cycle's

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

penalty. This isn't a problem for the SID or generally for the CIAs, but the timing of the video acquire routine is so tight this will make it occasionally glitch, even with its delay constants adjusted for the faster clock.

More importantly, however, it turns out most of the time spent in acquisition is in the busy loops waiting for the next sync. Double-speed isn't going to fix that and isn't worth making the output iffy as a result. The amount of time to acquire the screen is a function of how quickly the ComputerEyes can scan it, which remains a constant no matter how quickly the 128 can read and assemble an image from it.

The first order of business is to make a VDC graphics mode as close as possible to the VIC's 320x200 high-resolution bitmap display, because we want to do as little post-processing as possible, and we certainly don't want to implement an expensive pixel-doubling routine. Yes, while the rudimentary 128 demoscene tries to squeeze out more pixels and more colours, we're going to bring you *fewer* pixels and *two* colours. Drumroll please — allow me to present a *low* resolution VDC display:



All the assembly and BASIC source code we'll be using in this entry is available on Github (see below). This small demonstration of low-resolution VDC graphics is in the test/ folder.

You'll need Perl 5 and the xa65 cross-assembler to build it; the Perl linker tool to glue it all together into a runnable file is included in the Github project. It only requires 16K VDC RAM, runs on NTSC, and should run on PAL, or at least VICE will run it in PAL, anyway. The image is video-standard compliant, and any composite monitor will work with the resulting signal from the 128's RGB port pin or with any compatible RGB monitor like the 1902 or 1084. Build the project with `make` and `LOAD` and `RUN` the resulting program *in 64 mode*. It will display the bitmap on the VIC screen, copy it to the VDC screen, and then cycle the VDC colours. With VICE, you can see both displays at once.

This display works by using the VDC's built-in ability to double its pixels. To match the resolution we also halve the total and displayed numbers of horizontal character positions and slightly increase the character width, all of which are standard VDC register features. For example, here's a small BASIC program (minimally modified from COMPUTE!'s *Mapping the 128*, page 451) that makes a 40-column text display on the VDC in 128 mode:

```
10 wr=dec("cdcc"):rr=dec("cdda")
20 syswr,63,0:syswr,40,1:syswr,55,2
30 sysrr,,25:rrega:syswr,(aor16),25
40 syswr,137,22:syswr,40,27
50
poke238,39:printchr$(19)chr$(19)chr$(147)
)
```

This uses the built-in ROM routines to set and read VDC registers instead of BASIC `POKE` and `PEEK()`, which are implemented using indirect addressing and are therefore unsuitable. It also has the VDC skip 40 bytes between screen lines to get everything to align properly, and adjusts the horizontal sync position to the right to better situate it on the screen. The dance with reading register 25 before setting its double clock bit is due to the very earliest VDC chips handling

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

horizontal fine scrolling differently, which is part of the same register and must be preserved.

For high-res mode, though, there's a little more work to do. The VIC-II's bitmap mode is still essentially displaying 8x8 characters, just from a larger matrix instead of glyphs from a character set, but the VDC bitmap is organized as line-by-line runs of packed pixel data more like modern framebuffer. (In fact, it's almost exactly the same as a Netpbm P4 PBM image, something we'll make use of.) We really want to take advantage of the VDC's auto-increment feature for pushing data into it without having to repeatedly change its address registers, so we do as much work as we can to send the VDC linear data and do any address hopscotching in the VIC's bitmap instead.

To set up the display is also slightly more complicated. First, we initialize the VDC registers like the 128 Kernal would have since it may not have necessarily happened if we go straight to 64 mode, including the undocumented 38th register in the 8568 to ensure positive polarity for hsync and vsync (adjust this if you want to use an EGA or VGA monitor). VICE's VDC emulation is not exact, so for testing purposes I needed to find a sequence of register stores that worked on both VICE emulating a PAL flat 128 and on my real NTSC 128DCR. The working way is to turn on VDC high-res mode, adjust the two horizontal character position numbers, then turn on the double pixel clock and turn off individual attributes, set the character width, and set the horizontal sync position using a slightly different value. This generated a good image on both VICE and my real 128DCR, but the real machine also showed an unusual defect:



See that "dot crawl" on the right edge of the displayed area? It shimmers with data noise and is quite distracting since the rest of the image is perfectly stable, but is less apparent if there are no pixels at the beginning of the next row. After I unsuccessfully tried covering it up with horizontal blanking or playing with attributes to mask it, the working solution was to stuff a dummy null byte at the end of a row of pixels and tell the VDC to skip it. Since the noise was dependent on the next byte the VDC fetched, giving it a null meant no pixels to display, and the noise disappeared. VICE doesn't seem to model this. Because we're still just displaying 40 horizontal character positions (i.e., 320 pixels), what would effectively be the 41st character position is fetched and never displayed, leaving a stable image and bloating the bitmap data by only 200 additional bytes.

(Incidentally, there are other practical uses for this display mode. It's loading roughly half the data it would have for a full display, so it paints faster and we don't have to wait on it as much, and it gives you the exact same resolution as the VIC-II but with all the things the VDC is good at, namely scrolling, blitting, non-interference, and 2MHz support. More importantly, the bitmap's reduced memory usage compared to the 640x200 default makes it compact enough to support hi-res with attributes even using a 16K VDC. If you don't need sprites or multicolour mode, this particular configuration could be a compelling alternative.)

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

With that, we're ready to create our VDC live view, so now we'll need the ComputerEyes driver. In an era where driver software wasn't necessarily considered a freely distributable commodity, Digital Vision was very forward-thinking; the more applications out there that took advantage of the hardware, the more of them they figured they'd sell, so they included the assembler source of the driver (the BASIC program that provided the main menu was just a BASIC program) and even had a specific section in the manual on how to copy and use the driver in your own programs. The driver includes code to manually calibrate the device's sync and brightness, capture at multiple levels, and compress and decompress hi-res images. A jump table made it straightforward to use and the manual included all of the addresses to call from BASIC or machine language.

In our Github project, I provide the same ComputerEyes assembler source, just converted to build with `xa65` (I've tried to keep it the same otherwise, with the same comments), as well as the BASIC menu program as ASCII text. Hope they don't mind. When you build the project with `make`, the BASIC text is tokenized and the driver is assembled (the resulting machine language binary is bitwise identical with the original), and then they are combined with a directory utility also on the ComputerEyes disk to yield a one-part file named `ced` that you can simply `LOAD` and `RUN`. The Perl tools to tokenize and link the project are included. This one-part version is functionally the same as what you would have gotten on the ComputerEyes floppy, which had the same components as separate files the BASIC menu loaded from disk instead. Please note that the help files are not included; you can get a .d64 image of the original disk [at] <https://gmontag451.wordpress.com/2015/02/15/computereyes-on-the-commodore-64/>



Time to plug in our trusty little Sharp NTSC CCD camera. We're going to use nearly the same code we used in the 320x200 demo, because we're doing nearly the same thing, i.e., repeatedly linearizing a VIC hi-res bitmap to VDC RAM. Using the jump table, we'll first call the routines to calibrate the sync and brightness, then start an endless loop of "fast" captures (what the manual calls "normal acquire") and copying the resulting image to VDC memory (this is done at 2MHz and returned to 1MHz after). We'll also add code to not do the copy if `SHIFT LOCK` is down so if you get an image you like, you can freeze it.

And, of course, if you've frozen an image you may want to save it, too. Since the linear bitmap format of the VDC is nearly exactly a P4 Portable Bit Map image (we just have to add a header and invert the bytes), if you have the image frozen with `SHIFT LOCK`, you can additionally hold down the Commodore key until the disk starts up to save the current image as a PBM file to the current disk drive. It will save grabs sequentially as `0VDCGRAB.PBM`, `1VDCGRAB.PBM`, etc. (any files with those names will be deleted and overwritten), skipping each scanline's null gap byte automatically. You can then transfer the grab to your regular desktop computer as most tools can convert or open a PBM image. Here's one of me acquired exactly thus:

**Scientist added the letters AH to the periodic table. It is the element of surprise.**



Building the project with make also creates two additional one-part programs that do all this and also LOAD and RUN in 64 mode, namely vdcfast and vdcslow. vdcslow is what we just described above, politely calling the documented entry points in the driver. This reliably works. We have our live view, every, uh, seven seconds!

But ... as the name implies, we can do it faster!

We've already just about conclusively established that the actual acquisition step can't be flogged to go any faster because it's limited by the hardware. If there are any savings to be had, they'll have to be in the overhead, conclusion or initialization of the normal "fast" acquisition routine. The code to integrate each vertical pixel line into the VIC bitmap isn't perfectly efficient, but it runs between hsyncs, so making it marginally faster would just make it "hurry up and wait" for the next sync, yielding no gain. At the last column, it pretty much just turns the ComputerEyes hardware capture off and exits, so we won't get much there either.

That leaves the initialization. Here's the routine NACQ which our "slow" driver calls:

```
*
* NORMAL ACQUISITION
*
NACQ LDA #7 ;LEVEL = 7
    STA THRESH
PACQ LDA #$EF ;BLANK SCREEN
    JSR INIT ;INITIALIZE THINGS
    JSR COLOR ;INIT SCREEN COLOR
    LDA #DCOMP ;SET DELAY COMP VALUE
```

```
STA DCOMPV
JSR ACQ ;ACQUIRE
*
PFINI LDA #$10 ;UNBLANK SCREEN
    ORA VCR
    STA VCR
    JSR SWAP ;RESTORE ZPAGE LOCS
    CLI ;ENABLE INTERRUPTS
    RTS
```

INIT and SWAP look like this:

```
*
* INITIALIZATION SUBROUTINE
*
INIT AND VCR ;CONDITIONALLY BLANK SCREEN
    STA VCR
*
INITB LDA VMCR ;PUT BIT MAP @ LOC $2000
    ORA #$08
    STA VMCR
*
    LDA VCR ;ENTER BIT MAP MODE
    ORA #$20
    STA VCR
*
INITA LDA #$3F ;SET DATA DIRECTION REG
    TO
    STA DIRREG ; 00111111 (1 = OUTPUT)
*
    SEI ;DISABLE INTERRUPTS
*
    JSR SWAP ;FREE UP ZPAGE LOCS
    RTS
*
[...]
```

```
* MEMORY SWAP ROUTINE FOR PACK/UNPACK
*
SWAP LDX #$0C ;SWAP LOCS 2 THRU $0E
SWAPLP LDA $02,X ;STORE ON STACK
    PHA
    LDA TBUF,X ;MOVE FROM TBUF TO ZPAGE
    STA $02,X
    PLA
    STA TBUF,X ;STORE ZPAGE IN TBUF
    DEX ;NEXT LOC
    BPL SWAPLP
*
    RTS ;DONE
```

Swapping 12 bytes with the processor stack in this fashion isn't a particularly quick way to do it (PLA and PHA are comparatively expensive instructions, another reason the 6502 is unusually

**Scientist added the letters AH to the periodic table. It is the element of surprise.**



hostile to stack-based languages), but the routine runs exactly twice in the entire capture process, so its wallclock contribution is negligible. The other pieces are single stores. We can eliminate the call to JSR COLOR (which just sets up the screen memory for hi-res attributes), because we've already done it and the VDC doesn't need it, but that's also a trivial portion of the runtime.

We then move on into ACQ, the meat of the acquisition subroutine. This is not intended as an external-callable function. The beginning looks like this (I included the comment at the end so you can match it up with the assembler source back at the beginning):

```
*
*           MAIN ACQUISITION SUBROUTINE
*
* THE TIMING OF THESE LOOPS IS VERY
CRITICAL.
* DO NOT MODIFY, OR AT LEAST BE VERY
CAREFUL!
*
ACQ LDA THRESH ;SET THRESHOLD AND START
  ORA #$08 ; VIDEO INTERFACE
  STA CEREG
*
  LDA #0 ;INIT X OFFSET REGISTER
  STA XOFF
  STA XOFF+1
*
* WAIT NVERT VERT SYNCs BEFORE BEGINNING
ACQUISITION
*
  LDA #NVERT ;DELAY COMPENSATION FOR
  CLC ; FIRST SCAN
  ADC DCOMPV
  TAY
WVLP JSR WVVERT ;CALL WAIT-FOR-VERT
ROUTINE
  DEY
  BNE WVLP ;DO THIS NVERT TIMES
*
* NOW BEGIN ACQUIRING - INIT BIT
POSITION LOCATION
*
MLP LDA #$80 ;BIT 7 IS LEFT-MOST BIT ON
SCREEN
  STA BITPOS
*
```

```
* ACQUIRE ONE COLUMN - WAIT FOR VSYNC
PLUS SOME HSYNC
*
```

There is an apparently innocent comment saying to "wait NVERT syncs before beginning acquisition." That quantity is further added to a variable DCOMPV to yield the number of vertical syncs we wait for before getting to it. If you look back at the beginning of NACQ, it sets that variable. How many vertical syncs does it end up waiting for?

```
NVERT EQU 76
THRESH EQU $CFF0
DCOMPV EQU $CFF1 ;DELAY COMPENSATION REG
DCOMP EQU 9 ;DELAY COMPENSATION VALUE
```

$76+9=85$  vertical syncs means we're waiting for 85 *entire frames* to get started. At 16.68 milliseconds per frame (59.94fps), that's almost a full second and a half before we even grab our first sample!

The reason DCOMPV is a variable is it isn't strictly necessary to wait the 9 frames if you immediately go back and capture again. In fact, the grey-scale routine sets that count to zero after the first capture is acquired and only waits 76 frames for the other captures thereafter. But dispensing with those nine only buys us around a seventh of a second. What happens if we put all that initialization into our own code, then start up the ComputerEyes by setting bit 3 and jump into the middle of the acquisition routine after ... just one vertical sync?

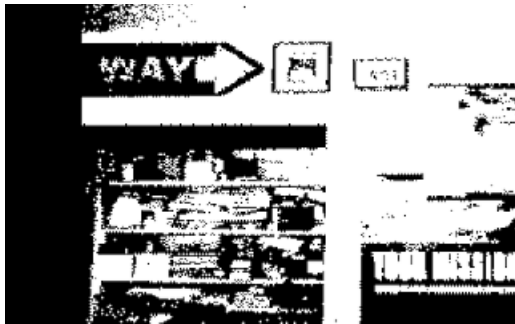


**Scientist added the letters AH to the periodic table. It is the element of surprise.**

Compare the view on the screen with the room (orient yourself using the ONE WAY sign). The first 10% or so of the screen is an echoed strip out from near the middle of it, but the rest of the screen is acquired normally and in the right order. In fact, it's a precise, constant 34 samples out of 320. Although we've lost that part of the image, the part that we lose seems to be consistent and predictable on both my older and newer ComputerEyes, and also no matter at which point I switched from the brightness calibrator (which just calls the same routine, but with the screen on), so I'm fairly certain this is a reliable feature of the hardware instead of merely a quirk of my devices.

Given that we do need to wait at least *one* vsync to ensure we're capturing from the top, this one ~~weird~~ change puts us nearly at our theoretical minimum capture time -- 5,334 milliseconds for the screen including the blown 34 samples, plus an additional frame in the worst case if we hit it just after the previous vsync, to yield no more than 5.5 seconds. That matches well with what I get from a stopwatch measuring from VDC paint to VDC paint.

The simplest way to deal with that is just to black out those 34 leftmost samples when we copy to VDC RAM. That's what we do in `vdcfast`, and here's the result:



I'm not completely certain of what we're doing to the hardware here, but my best guess is the same process that inches the samples across the screen each frame needs some additional number of frames to get back to the starting point, and that

time is also largely constant. By short-circuiting that, we lose 11% of the screen but gain somewhere between 22 and 24% faster capture.

*[I asked Dave Pratt himself about this and he explained: "The reason for the 76(+9) delay is so that the samples are taken from the center of the image horizontally. If you skip the delay, the image isn't centered, but if that's OK with you, then no harm done."]*

Is that a fair trade? Watch this 5x time lapse video and judge for yourself, comparing my wife's motions with what you get on the screen (sorry about the portrait orientation but it was the quickest way to get everything in frame):

Don't tell the Skynet Hunter-Killers about this or they're all gonna want one.

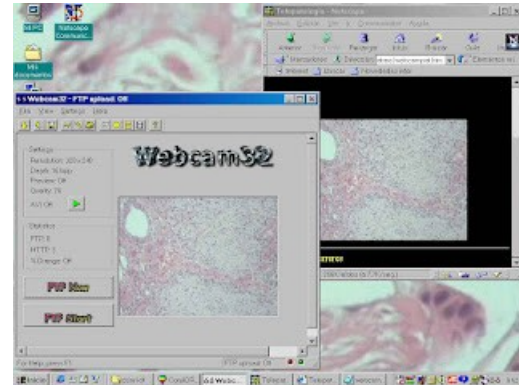


Curiously, there never appears to have been an Amiga ComputerEyes, but Digital Vision did produce a colour model for the Atari ST. The Macintosh and IBM PC-compatible lines, however, had their own distinct lineages from the original slow-scan hardware through to actual framebuffers. The initial \$250 [in 2023 about \$650] Mac ComputerEyes in 1998 was another external slow-scan device that connected over the mini-DIN serial ports and provided 256 levels of grey at up to 640x480, but the ComputerEyes/Pro was an actual 24-bit colour-capable NuBus slow-scan card, and the ComputerEyes/RT was a true external framegrabber connected via SCSI that pulled a entire 24-bit 640x480 image in a thirtieth

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

of a second and exported directly to QuickTime movies. That's it in the picture above, blown up from an Internet Archive copy of the corporate website. Instead of the earlier slow-scan hardware, the /RT ("Realtime") used off-the-shelf Brooktree Bt208 A/D converters for the RGB conversion and sold for \$600 (about \$1500 today). Finally, the technologically unrelated TelevEyes product generated a composite PAL or NTSC signal from the Mac's RGB output, practically the ComputerEyes in reverse. The Pro version even supported overlays on live video.

box, and the last of the line and its only PCI-based product, the ComputerEyes/PCI based on the Philips SAA7111 "VIP" (Video Input Processor). The PC also had its own TelevEyes product, and as you would expect, the PC products quickly became Digital Vision's biggest sellers.



One of the most interesting uses of the CE/PCI that I ran into doing the research for this article was the 1997 medical telepathology setup shown above.

<https://www.mcmaster.ca/inabis98/rojo/alfaro0464/two.html>

Digital Vision also sold videoconferencing packages for the Mac using the free CU-SeeMe tool, shown here in a news item from March 1995. The VDIG was freely downloadable if you already had the hardware.

It used a microscope, camera, CE/PCI, Microsoft Personal Web Server and off-the-shelf webcam software to transmit 640x480 tissue sample images over a 33.6Kbps modem for remote viewing at about two frames per second. (Hey, we're not *too* far off!)

The PC line was even more varied. It had its own slow-scan ComputerEyes (64 greys) and ComputerEyes/Pro (256 greys or 24-bit colour) ISA cards, though these were never external boxes, and its own ISA card version of the ComputerEyes/RT framegrabber capable of the same high-speed capture. But the line continued with the \$600 [\$1200] ComputerEyes/1024 in 1995, a Windows-compatible ISA card capable of 1024x512 capture at 24 bits using a Brooktree Bt254 digitizer, the \$400 [\$800] ComputerEyes/LPT developed in Taiwan which was basically the /RT in an external printer port

In 1995 Dave and Vi Pratt, still the company owners, sold the company after over a decade and the post-/RT products were launched under the new management. Unfortunately, the later administration was unable to continue the company's earlier success and it was acquired and dissolved by rival peripherals company Focus Enhancements in October 1996. The British stock photos company and subsequent Swedish video and multimedia company have no relation. Focus itself was acquired by French multimedia company VITEC in 2010, who is still in business

**Scientist added the letters AH to the periodic table. It is the element of surprise.**

and presumably remains the current owner of the Digital Vision IP.

Fortunately Dave Pratt put up his collected memories, documents, and photographs on virtually the entire Digital Vision line as an enduring memorial to his former company. All of the products mentioned here appear there, including photos, prototypes and product circulars. Go to

<http://www.digital-vision-inc.com/index.htm>

The source code for the ComputerEyes driver, our two VDC live capture tools and the 320x200 VDC demonstration are on Github at

<https://github.com/classilla/cepotato>

Meeting place reservation ..... Dick Estel

**-The Small Print-**

The Fresno Commodore User Group is a club whose members share an interest in Commodore 8-bit and Amiga computers. Our mailing address is 185 W. Pilgrim Lane, Clovis, CA 93612. We meet monthly in the meeting room of Panera Bread, 3590 West Shaw, Fresno, CA. The meetings generally include demonstrations, discussion, and individual help.

Dues are \$12 for 12 months. New members receive a "New Member Disk" containing a number of useful Commodore 8-bit utilities. Members receive a subscription to The Interface newsletter, access to the public domain disk library, technical assistance, and reduced prices on selected software/hardware.

Permission to reproduce content of The Interface is granted provided credit is given to the source, and when identified, the author. Club members are encouraged to submit articles, tips, or ideas for articles.

**Disclaimer** – The club, its officers, members, and authors are not responsible for the accuracy of the contents of The Interface or the results of actions based on its contents.

Our disk library contains over 3,000 public domain programs for the C64 and C128. Members are entitled to copies of these disks at no cost if a blank disk is provided. We do not deal with pirated, copyrighted, violent, or obscene programs. Please call our attention to any programs found in our library which may violate these standards.

# Club Officers

## >>--> Officers and Keypersons <--<<

President .....	Robert Bernardo
Vice-president .....	Roger Van Pelt
Secretary/Treasurer .....	Dick Estel
The Interface Editor .....	Lenard Roach
Librarian .....	Roger Van Pelt
Club equipment .....	Roger Van Pelt

**Scientist added the letters AH to the periodic table. It is the element of surprise.**